

Computation of vector sublattices and minimal lattice-subspaces of \mathbb{R}^k . Applications in finance.

V.N. Katsikis ^{a,*} I.A. Polyrakis ^b

^a*General Department of Mathematics, Technological Education Institute of Piraeus,
12244, Athens, Greece*

^b*Department of Mathematics, National Technical University of Athens, Zographou 15780,
Athens, Greece*

Abstract

In this article we perform a computational study of Polyrakis algorithms presented in [12,13]. These algorithms are used for the determination of the vector sublattice and the minimal lattice-subspace generated by a finite set of positive vectors of \mathbb{R}^k . The study demonstrates that our findings can be very useful in the field of Economics, especially in completion by options of security markets and portfolio insurance.

Key words: computational methods, minimal lattice-subspaces, vector sublattices, portfolio insurance, completion of security markets.

2000 Mathematics Subject Classification: 46N10;91B28

1 Introduction

This paper provides computational methods for the determination of the vector sublattice and the minimal lattice-subspace of \mathbb{R}^k generated by a set $B = \{x_1, x_2, \dots, x_n\}$ of linearly independent positive vectors of \mathbb{R}^k . In order to reach our goal the study of a vector-valued function β is further involved, which is defined and studied by I. Polyrakis in [12,13]. In particular, in [12,13], it is proved that if $B = \{x_1, x_2, \dots, x_n\}$ is a set of linearly independent positive vectors of \mathbb{R}^k and $\beta(i) = \frac{r(i)}{\|r(i)\|_1}$ for each $i = 1, 2, \dots, k$, where $r(i) = (x_1(i), x_2(i), \dots, x_n(i))$, $R(\beta)$ denotes the range of the function β and K is the convex hull of $R(\beta)$ then it holds,

* Vasilios N. Katsikis

Email addresses: vaskats@gmail.com (V.N. Katsikis), ypoly@math.ntua.gr (I.A. Polyrakis).

- (i) There exists an n —dimensional minimal lattice-subspace of \mathbb{R}^k containing B ,
- (ii) The vector sublattice generated by B is an m —dimensional subspace of \mathbb{R}^k .

where n is the number of vertices of K and m is the number of different values of β .

In [10], the vector sublattice generated by a set of linearly independent positive vectors of \mathbb{R}^k was used in order to provide an answer to the challenging question of market completion by options, in a discrete-time framework. Also, in [11] it is proved that if for a set of linearly independent positive vectors of \mathbb{R}^k there exists a minimal lattice-subspace containing this set, then one can use the produced subspace for the study of a cost-minimization problem that is also known as the minimum cost portfolio insurance problem. From the computational point of view, several methods are presented in [7,8,9] in order to check whether a finite collection of linearly independent positive vectors(resp. functions) of \mathbb{R}^k (resp. $C[a, b]$) forms a lattice-subspace or a vector sublattice. In addition, these methods have been used for the calculation of the minimum-cost portfolio insurance. The main goals of this article are outlined as follows:

- (1) Exploit the theory of positive bases and derive new computational methods for the calculation of the completion by options of a two-period security market,
- (2) Calculate the minimal lattice-subspace generated by a finite collection of linearly independent positive vectors.

The proposed numerical solution, based upon previous works of the authors, provide the exact solution to the problem of calculating the market completion by options and the minimum-cost insured portfolio. Also, a closer look at Example 12 can tell us that a manual procedure in order to determine the completion of the given security market can easily be a prohibited task. In the economics models that we are working the input data, as for example the payoffs of the different securities, are standard for a long time period and so, by our methods we determine the exact solution for these problems.

In the present work, the numerical tasks have been performed using the high-end Matlab programming language. Specifically, the Matlab 7.4 (R2007b) environment [5,6] was used on an Intel(R) Pentium(R) Dual CPU T2310 @ 1.46 GHz 1.47 GHz 32-bit system with 2 GB of RAM memory running on the Windows Vista Home Premium Operating System.

2 Lattice-subspaces and vector sublattices of \mathbb{R}^k

2.1 Preliminaries

We first recall some definitions and notation from the vector lattice theory. Let $\mathbb{R}^k = \{x = (x(1), x(2), \dots, x(k)) | x(i) \in \mathbb{R}, \text{ for each } i\}$, where we view \mathbb{R}^k as an ordered space. The *pointwise order* relation in \mathbb{R}^k is defined by

$$x \leq y \text{ if and only if } x(i) \leq y(i), \text{ for each } i = 1, \dots, k.$$

The positive cone of \mathbb{R}^k is defined by $\mathbb{R}_+^k = \{x \in \mathbb{R}^k | x(i) \geq 0, \text{ for each } i\}$ and if we suppose that X is a vector subspace of \mathbb{R}^k then X ordered by the pointwise ordering is an *ordered subspace* of \mathbb{R}^k with positive cone $X_+ = X \cap \mathbb{R}_+^k$. A point $x \in \mathbb{R}^k$ is an *upper bound* (resp. *lower bound*) of a subset $S \subseteq \mathbb{R}^k$ if and only if $y \leq x$ (resp. $x \leq y$), for all $y \in S$. For a two-point set $S = \{x, y\}$, we denote by $x \vee y$ (resp. $x \wedge y$) the *supremum* of S i.e., its least upper bound (resp. the *infimum* of S i.e., its greatest lower bound). Thus, $x \vee y$ (resp. $x \wedge y$) is the componentwise maximum (resp. minimum) of x and y defined by

$$(x \vee y)(i) = \max\{x(i), y(i)\} \text{ and } (x \wedge y)(i) = \min\{x(i), y(i)\}, \text{ for all } i = 1, \dots, k.$$

An ordered subspace X of \mathbb{R}^k is a *lattice-subspace* of \mathbb{R}^k if it is a vector lattice in the induced ordering, i.e., for any two vectors $x, y \in X$ the supremum and the infimum of $\{x, y\}$ both exist in X . Note that the supremum and the infimum of the set $\{x, y\}$ are, in general, different in the subspace from the supremum and the infimum of this set in the initial space. An ordered subspace Z of \mathbb{R}^k is a *vector sublattice* or a *Riesz subspace* of \mathbb{R}^k if for any $x, y \in Z$ the supremum and the infimum of the set $\{x, y\}$ in \mathbb{R}^k belong to Z .

Assume that X is an ordered subspace of \mathbb{R}^k and $B = \{b_1, b_2, \dots, b_n\}$ is a basis for X . Then B is a *positive basis* of X if for each $x \in X$ it holds that x is positive if and only if its coefficients in the basis B are positive. In other words, B is a positive basis of X if the positive cone X_+ of X has the form,

$$X_+ = \{x = \sum_{i=1}^n \lambda_i b_i | \lambda_i \geq 0, \text{ for each } i\}.$$

Then, for any $x = \sum_{i=1}^n \lambda_i b_i$ and $y = \sum_{i=1}^n \mu_i b_i$ we have $x \leq y$ if and only if $\lambda_i \leq \mu_i$ for each $i = 1, 2, \dots, n$.

Each element b_i of the positive basis of X is an extremal point of X_+ thus a positive basis of X is unique in the sense of positive multiples. Recall that a nonzero element x_0 of X_+ is an *extremal point* of X_+ if, for any $x \in X, 0 \leq x \leq x_0$ implies $x = \lambda x_0$ for a real number λ . The existence of positive bases is not always ensured, but in the case where X is a vector sublattice of \mathbb{R}^k then X has always a positive

basis. Moreover, it holds that an ordered subspace of \mathbb{R}^k has a positive basis if and only if it is a lattice-subspace of \mathbb{R}^k . If $B = \{b_1, b_2, \dots, b_n\}$ is a positive basis for a lattice-subspace (or a vector sublattice) X then the lattice operations in X , namely $x \nabla y$ for the supremum and $x \triangle y$ for the infimum of the set $\{x, y\}$ in X , are given by

$$x \nabla y = \sum_{i=1}^n \max\{\lambda_i, \mu_i\} b_i \text{ and } x \triangle y = \sum_{i=1}^n \min\{\lambda_i, \mu_i\} b_i,$$

for each $x = \sum_{i=1}^n \lambda_i b_i, y = \sum_{i=1}^n \mu_i b_i \in X$. A vector sublattice is always a lattice-subspace, but the converse is not true as shown in the next example,

Example 1 Let $X = [x_1, x_2, x_3]$ be the subspace of \mathbb{R}^4 generated by the vectors $x_1 = (6, 0, 0, 1), x_2 = (6, 4, 0, 0), x_3 = (8, 4, 2, 0)$. An easy argument shows that the set $B = \{b_1, b_2, b_3\}$ where

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 2 & 0 \\ 12 & 8 & 0 & 0 \\ 6 & 0 & 0 & 1 \end{bmatrix}$$

forms a positive basis of X therefore X is a lattice-subspace of \mathbb{R}^4 . On the other hand, let us consider the vectors $y_1 = 2x_1 + x_2 = (18, 4, 0, 2)$ and $y_2 = x_3 - x_2 = (2, 0, 2, 0)$ of X . Then, $y_1 \vee y_2 = (18, 4, 2, 2)$ and since $y_1 = \frac{1}{2}b_2 + 2b_3, y_2 = b_1$, it follows that $y_1 \nabla y_2 = b_1 + \frac{1}{2}b_2 + 2b_3 = (20, 4, 4, 2)$. From the definition of a vector sublattice we have that X is a vector sublattice of \mathbb{R}^4 if for each $x, y \in X$ it holds $x \vee y = x \nabla y \in X$ and $x \wedge y = x \triangle y \in X$. Therefore, since there are two elements $y_1, y_2 \in X$ such that $y_1 \vee y_2 \neq y_1 \nabla y_2$ it follows that X is not a vector sublattice of \mathbb{R}^4 .

For an extensive presentation of lattice-subspaces, positive bases and vector sublattices, we refer to [1,11,12,13], for computational methods in positive bases theory we refer to [7,8,9] and for several applications in the theory of finance the reader may refer to [2,3,4,10].

2.2 The mathematical problem

In this section we shall present the results of [12,13] for the construction of a minimal lattice-subspace of \mathbb{R}^k containing a linearly independent subset of \mathbb{R}_+^k together with the construction of the vector sublattice of \mathbb{R}^k that this subset generates. In particular, given a collection of linearly independent, positive vectors, x_1, x_2, \dots, x_n of \mathbb{R}^k then the basic tool for our study is a special function, named basic function, that this collection of vectors defines. This function was first introduced and studied in [12]. In our analysis we shall use the notation introduced in [12], so let us denote

by r the function $r : \{1, 2, \dots, k\} \rightarrow \mathbb{R}^k$ such that

$$r(i) = (x_1(i), x_2(i), \dots, x_n(i))$$

and by β the function $\beta : \{1, 2, \dots, k\} \rightarrow \mathbb{R}^k$ such that

$$\beta(i) = \frac{r(i)}{\|r(i)\|_1}$$

for each $i \in \{1, 2, \dots, k\}$ with $\|r(i)\|_1 \neq 0$. We shall refer to β as the *basic function* of the vectors x_1, x_2, \dots, x_n . The set

$$R(\beta) = \{\beta(i) | i = 1, 2, \dots, k, \text{ with } \|r(i)\|_1 \neq 0\},$$

is the *range* of the basic function and the *cardinal number*, $\text{card}R(\beta)$, of $R(\beta)$ is the number of different elements of $R(\beta)$. Let $\text{card}R(\beta) = m$ then $n \leq m \leq k$ and by K we shall denote the convex hull of $R(\beta)$ which is, as the convex hull of a finite subset of \mathbb{R}^k , a polytope with d vertices and each vertex of K belongs to $R(\beta)$. It is clear that $n \leq d \leq m$.

An essential issue for our analysis are the conditions under which a collection of linearly independent, positive vectors x_1, x_2, \dots, x_n of \mathbb{R}^k can be used to derive a minimal lattice-subspace and a vector sublattice containing these vectors. In [13], theorem 3.19 is a criterion for lattice-subspaces and vector sublattices and provides a full answer on the topic on the basis of describing the geometry of this problem. In the case where $X = [x_1, x_2, \dots, x_n]$ is a lattice-subspace or a vector sublattice of \mathbb{R}^k then the theorem determines a positive basis in X while, in the opposite case, theorem 3.19 provides a minimal lattice-subspace and a vector sublattice containing X . In order to state theorem 3.19, let us consider $R(\beta) = \{P_1, P_2, \dots, P_m\}$ such that the first n vertices P_1, P_2, \dots, P_n are linearly independent and P_1, P_2, \dots, P_d are the vertices of K , $n \leq d \leq m$. Also, A^T denotes the transpose matrix of a matrix A .

Theorem 2 [13, Theorem 3.19]. *Suppose that the above assumptions are satisfied. Then,*

- (i) *X is a vector sublattice of \mathbb{R}^k if and only if $R(\beta)$ has exactly n points (i.e., $m = n$). Then a positive basis b_1, b_2, \dots, b_n for X is defined by the formula*

$$(b_1, b_2, \dots, b_n)^T = A^{-1}(x_1, x_2, \dots, x_n)^T,$$

where A is the $n \times n$ matrix whose i th column is the vector P_i , for each $i = 1, 2, \dots, m$.

- (ii) *X is a lattice-subspace of \mathbb{R}^k if and only if the polytope K has n vertices (i.e., $d = n$). Then a positive basis b_1, b_2, \dots, b_n for X is defined by the formula*

$$(b_1, b_2, \dots, b_n)^T = A^{-1}(x_1, x_2, \dots, x_n)^T,$$

where A is the $n \times n$ matrix whose i th column is the vector P_i , for each $i = 1, 2, \dots, d$.

(iii) Let $m > n$. If $I_s = \beta^{-1}(P_s)$, and

$$x_s = \sum_{i \in I_s} \|r(i)\|_1 e_i, \quad s = n+1, n+2, \dots, m,$$

then

$$Z = [x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_m]$$

is the vector sublattice generated by x_1, x_2, \dots, x_n and $\dim Z = m$.

(iv) Let $d > n$. If $\xi_i : D(\beta) \rightarrow \mathbb{R}_+, i = 1, 2, \dots, d$ such that $\sum_{i=1}^d \xi_i(j) = 1$ and $\beta(j) = \sum_{i=1}^d \xi_i(j) P_i$ for each $j \in D(\beta)$, and $x_{n+i}, i = 1, 2, \dots, d - n$, are the following vectors of \mathbb{R}^k :

$$x_{n+i} = \sum_{j \in D(\beta)} \xi_{n+i}(j) \|r(j)\|_1 e_j,$$

then

$$Y = [x_1, \dots, x_n, x_{n+1}, \dots, x_d]$$

is a minimal lattice-subspace of \mathbb{R}^k containing x_1, x_2, \dots, x_n and $\dim Y = d$.

In [9], a new numerical package is provided so that, for a variety of dimensions and subspaces, we are able to check conditions (i), (ii) of Theorem 2. In addition, in the case where $X = [x_1, x_2, \dots, x_n]$ is a vector sublattice or a lattice-subspace the proposed method, in [9], provides the positive basis of X therefore a complete description of the vector sublattice or the lattice-subspace is given.

In the case where X is not a vector sublattice or a lattice-subspace, an essential issue for our analysis is the construction of a powerful and efficient package in order to calculate the vector sublattice and a minimal lattice-subspace containing X by using conditions (iii), (iv) of Theorem 2.

2.3 The algorithm

- (1) Determine the function β as well as the range, $R(\beta)$, of β .
- (2) Compute the number $m = \text{card} R(\beta)$, and the number d of vertices of the polytope K .
- (3) If $n = m$ (vector sublattice case) or $n = d$ (lattice-subspace case) then, determine a positive basis of X .
- (4) If $m > n$, then $Z = [x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_m]$ is the vector sublattice generated by x_1, x_2, \dots, x_n , where $x_s = \sum_{i \in I_s} \|r(i)\|_1 e_i$ and $I_s = \beta^{-1}(P_s)$ for each $s = n+1, n+2, \dots, m$.
- (5) If $d > n$, then $Y = [x_1, \dots, x_n, x_{n+1}, \dots, x_d]$ is a minimal lattice-subspace of \mathbb{R}^k containing x_1, x_2, \dots, x_n , where the vectors $x_{n+i} = \sum_{j \in D(\beta)} \xi_{n+i}(j) \|r(j)\|_1 e_j$, $i = 1, \dots, d - n$, were defined in (iv) of Theorem 2.

Note that, the steps of this algorithm are based upon Theorem 2. In the following section, we present the translation followed by the implementation of this algorithm in \mathbb{R}^k within two Matlab-based functions named `SUBlat` and `MINlat`. These functions, as we shall see in Subsection 3.1, provides an important tool in order to investigate minimal lattice-subspaces and vector sublattices of \mathbb{R}^k generated by a set of linearly independent, positive vectors x_1, x_2, \dots, x_n of \mathbb{R}^k .

3 The computational method

3.1 Method presentation and examples

Our proposed numerical method is based on the introduction of two functions, namely `SUBlat` and `MINlat` (see Appendix), that enable us to perform fast testing for a variety of dimensions and subspaces. Also, both of these functions are using the function `SUBlatSUB` from [9], in order to calculate a positive basis. For the sake of completeness, we provide the `SUBlatSUB` function in the Appendix. Recall that, the numbers n, m, d, k denote the dimension of X , the cardinality of $R(\beta)$, the number of vertices of the convex hull of $R(\beta)$ and the dimension of the initial Euclidean space, respectively.

The function `SUBlat` first determines $R(\beta)$ and the number $m = \text{card}R(\beta)$ and then the n linearly independent vertices $P_i, i = 1, \dots, n$ of the polytope K . Finally, the program calculates the subsets $I_s = \beta^{-1}(P_s)$ and then the vectors

$$x_s = \sum_{i \in I_s} \|r(i)\|_1 e_i,$$

for each $s = n + 1, n + 2, \dots, m$. Therefore, since by theorem 2 (iii) the subspace

$$Z = [x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_m]$$

is the vector sublattice generated by the given vectors x_1, x_2, \dots, x_n the program responds, by using the function `SUBlatSUB` of [9], with two $m \times k$ matrices. The rows of the first matrix are the vectors

$$x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_m,$$

while the rows of the second matrix are the vectors b_1, \dots, b_m of the positive basis for the subspace Z .

In order to determine a minimal lattice-subspace generated by the given collection of vectors we use the function `MINlat`. The `MINlat` function first determines $R(\beta)$ and then the vertices of the polytope K . The correct performance of the `MINlat` function requires the use of the **convhulln** Matlab function which is

based on Qhull (For information about Qhull see <http://www.qhull.org/>). In order to determine a minimal lattice-subspace of \mathbb{R}^k containing x_1, x_2, \dots, x_n the program calculates the vectors $\xi_i : D(\beta) \rightarrow \mathbb{R}_+, i = 1, 2, \dots, d$ by solving k underdetermined $n \times d$ linear systems subject to the inequalities $\xi_i \geq 0, i = 1, \dots, n$. Finally, the program defines the vectors,

$$x_{n+i} = \sum_{j \in D(\beta)} \xi_{n+i}(j) \|r(j)\|_1 e_j, \quad i = 1, 2, \dots, d - n,$$

and since the subspace

$$Y = [x_1, \dots, x_n, x_{n+1}, \dots, x_d]$$

is a minimal lattice-subspace containing x_1, x_2, \dots, x_n the program provides, by using the function `SUBlatSUB` of [9], two $d \times k$ matrices. The rows of the first matrix are the vectors

$$x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_d,$$

while the rows of the second matrix are the vectors b_1, b_2, \dots, b_d of the positive basis for the subspace Y .

Consequently, let x_1, x_2, \dots, x_n be a collection of linearly independent, positive vectors of \mathbb{R}^k , then we construct a matrix B whose columns are the vectors of the given collection and then we apply the functions `SUBlat` and `MINlat` on that matrix as follows,

```
[VectorSublattice, Positivebasis]=SUBlat(B)
[Minimallatticesubspace, Positivebasis]=MINlat(B)
```

In order to illustrate the most important features of `SUBlat` and `MINlat`, we reproduce two examples featured in [7,13].

Example 3 Consider the following 10 vectors x_1, x_2, \dots, x_{10} in \mathbb{R}^{17} ,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 3 & 4 & 1 & 10 & 11 & 60 & 0 & 12 & 4 & 32 & 13 \\ 2 & 1 & 40 & 30 & 2 & 23 & 4 & 5 & 2 & 9 & 12 & 1 & 1 & 1 & 5 & 33 & 14 \\ 3 & 10 & 20 & 10 & 3 & 24 & 5 & 6 & 3 & 8 & 13 & 2 & 2 & 2 & 6 & 34 & 15 \\ 4 & 30 & 30 & 0 & 4 & 25 & 6 & 7 & 0 & 7 & 14 & 3 & 3 & 3 & 7 & 35 & 16 \\ 5 & 40 & 1 & 0 & 5 & 0 & 7 & 8 & 5 & 6 & 15 & 0 & 0 & 3 & 8 & 0 & 17 \\ 6 & 50 & 0 & 0 & 6 & 27 & 8 & 9 & 0 & 5 & 16 & 50 & 11 & 12 & 5 & 37 & 18 \\ 7 & 1 & 1 & 1 & 7 & 28 & 9 & 10 & 0 & 4 & 17 & 0 & 40 & 5 & 4 & 38 & 19 \\ 8 & 3 & 0 & 0 & 1 & 29 & 10 & 11 & 1 & 3 & 18 & 10 & 0 & 10 & 3 & 39 & 20 \\ 9 & 50 & 0 & 40 & 2 & 30 & 11 & 12 & 12 & 2 & 19 & 0 & 10 & 10 & 2 & 40 & 21 \\ 10 & 70 & 40 & 1 & 3 & 31 & 12 & 13 & 13 & 1 & 20 & 2 & 2 & 2 & 1 & 41 & 22 \end{bmatrix}$$

In [7] it is provided that the subspace $X = [x_1, x_2, \dots, x_{10}]$ is not a lattice-subspace of \mathbb{R}^{17} . In order to calculate the vector sublattice that X generates as well as a minimal lattice-subspace containing X we invoke the SUBlat and the MINlat functions by typing in the command window of the Matlab environment:

```
>>[VectorSublattice,Positivebasis]=SUBlat(B)
>>[Minimallatticesubspace,Positivebasis]=MINlat(B)
```

where B denotes a matrix whose columns are the vectors x_1, x_2, \dots, x_{10} . The results, then, are as follows:

```
VectorSublattice =
1  1  0  0  1  0  3  4  1  10  11  60  0  12  4  32  13
2  1  40 30  2 23  4  5  2  9  12  1  1  1  5  33  14
3  10 20 10  3 24  5  6  3  8  13  2  2  2  6  34  15
4  30 30  0  4 25  6  7  0  7  14  3  3  3  7  35  16
5  40  1  0  5  0  7  8  5  6  15  0  0  3  8  0  17
6  50  0  0  6 27  8  9  0  5  16 50 11 12  5  37  18
7  1  1  1  7 28  9 10  0  4  17  0 40  5  4  38  19
8  3  0  0  1 29 10 11  1  3  18 10  0 10  3  39  20
9  50  0 40  2 30 11 12 12  2 19  0 10 10  2  40  21
10 70 40  1  3 31 12 13 13  1 20  2  2  2  1  41  22
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  175
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  329  0
0  0  0  0  0  0  0  0  0  0  0  0  0  60  0  0  0
0  0  0  0  0  0  0  0  0  0  0 128  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0 155  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0 55  0  0  0  0  0  0  0
0  0  0  0  0  0  0 85  0  0  0  0  0  0  0  0  0

Positivebasis =
0  0  0  0  0  0  0  0  0  0  0  0  69  0  0  0  0
0  0  0  0  0 217  0  0  0  0  0  0  0  0  0  0  0
0  0 132  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0 82  0  0  0  0  0  0  0  0  0  0  0  0  0
0 256  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```

55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 170 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 37 0 0 0 0 0 0 0
0 0 0 0 34 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 310 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 350
0 0 0 0 0 0 75 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 658 0
0 0 0 0 0 0 0 0 0 0 0 0 45 0 0
0 0 0 0 0 0 0 0 110 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 120 0 0 0
0 0 0 0 0 0 0 0 0 256 0 0 0 0 0

```

Minimallatticesubspace =

```

1 1 0 0 1 0 3 4 1 10 11 60 0 12 4 32 13
2 1 40 30 2 23 4 5 2 9 12 1 1 5 33 14
3 10 20 10 3 24 5 6 3 8 13 2 2 6 34 15
4 30 30 0 4 25 6 7 0 7 14 3 3 3 7 35 16
5 40 1 0 5 0 7 8 5 6 15 0 0 3 8 0 17
6 50 0 0 6 27 8 9 0 5 16 50 11 12 5 37 18
7 1 1 1 7 28 9 10 0 4 17 0 40 5 4 38 19
8 3 0 0 1 29 10 11 1 3 18 10 0 10 3 39 20
9 50 0 40 2 30 11 12 12 2 19 0 10 10 2 40 21
10 70 40 1 3 31 12 13 13 1 20 2 2 2 1 41 22
0 0 0 0 0 0 10 15 0 55 50 0 0 0 0 60
0 0 0 0 0 0 0 0 0 0 0 0 60 0 0 0
0 0 0 0 0 0 0 0 0 0 128 0 0 0 0 0

```

Positivebasis =

```

0 0 0 0 0 0 0 0 0 0 0 0 69 0 0 0 0
0 0 0 0 0 217 0 0 0 0 0 0 0 0 0 0 0
0 0 132 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 82 0 0 0 0 0 0 0 0 0 0 0 0
0 256 0 0 0 0 0 0 0 0 0 0 0 0 0 0
55 0 0 0 0 0 65 70 0 0 105 0 0 0 0 115
0 0 0 0 0 0 0 0 37 0 0 0 0 0 0 0
0 0 0 0 34 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 45 0 0
0 0 0 0 0 0 20 30 0 110 100 0 0 0 0 120
0 0 0 0 0 0 0 0 0 0 0 0 0 0 329 0
0 0 0 0 0 0 0 0 0 0 0 0 120 0 0 0
0 0 0 0 0 0 0 0 0 0 256 0 0 0 0 0

```

Example 4 Consider the following 4 vectors x_1, x_2, x_3, x_4 in \mathbb{R}^7 ,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 0 & 1 & 1 & 4 \\ 0 & 1 & 1 & 1 & 1 & 0 & 2 \\ 2 & 1 & 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

where following the same procedure, as before, one gets

VectorSublattice =

```

1 2 1 0 1 1 4
0 1 1 1 1 0 2
2 1 0 1 1 1 2
1 0 1 1 1 0 0
0 0 0 0 0 2 0
0 4 0 0 0 0 8

```

```

Positivebasis =
0      0      0      3      0      0      0
0      0      0      0      0      4      0
4      0      0      0      0      0      0
0      8      0      0      0      0      16
0      0      0      0      4      0      0
0      0      3      0      0      0      0

Minimallatticesubspace =
1      2      1      0      1      1      4
0      1      1      1      1      0      2
2      1      0      1      1      1      2
1      0      1      1      1      0      0
0      4      0      0      0      0      8

Positivebasis =
0      0      0      3      1.5      0      0
0      8      0      0      0      0      16
4      0      0      0      0      0      0
0      0      3      0      1.5      0      0
0      0      0      0      1      2      0

```

Remark 5 In the beginning of the present section, we mentioned that in order to determine the vectors $\xi_i : D(\beta) \rightarrow \mathbb{R}_+, i = 1, 2, \dots, d$ the function `MINlat` responds by solving k underdetermined $n \times d$ linear systems subject to the inequalities $\xi_i \geq 0, i = 1, \dots, n$. Since, the solution to such systems is not unique then, if p denotes a solution corresponding to one of these systems, then $y = p + h$, where h is an arbitrary vector from the null space, is a solution too. Thus, a minimal lattice-subspace containing the vectors x_1, x_2, \dots, x_n is not unique. In [13], Example 3.21, it is proved that for the collection of vectors of Example 4 there exist two minimal lattice-subspaces Y, Y' such that

- $Y \neq Y'$,
- the space $Y \cap Y'$ is not a lattice-subspace,
- and Y, Y' are not subspaces of Z (Z denotes the generated vector sublattice).

3.2 Comparison Results

For the purpose of monitoring the performance, in this section we present the execution times of the proposed methods (`SUBlat`, `MINlat`) for various collections of vectors and dimensions. For this purpose we have used the function `testMINlat`, `testSUBlat` (see Appendix) in order to test 50 full rank matrices for each rank $n, n = 3, \dots, 30$. The cumulative results are presented in Table 1 while, the time responses have been recorded using the Matlab function **profile**.

It is evident, from Table 1, that the proposed numerical methods, based on the introduction of the `SUBlat` and the `MINlat` function, enable us to perform fast estimations for a variety of dimensions.

Table 1

Results for 50 tested full rank matrices for each rank $n, n = 3, \dots, 30$.

Rank	SUBlat	MINlat	Rank	SUBlat	MINlat
	(Total time in seconds)	(Total time in seconds)		(Total time in seconds)	(Total time in seconds)
3	0.553	0.689	17	2.812	1.528
4	0.569	0.705	18	2.994	1.732
5	0.584	0.742	19	3.333	1.885
6	0.696	0.822	20	3.594	2.222
7	0.856	0.831	21	4.084	2.428
8	0.973	0.890	22	4.219	2.773
9	1.141	0.903	23	4.569	3.175
10	1.307	1.012	24	4.817	3.695
11	1.642	1.047	25	5.149	4.306
12	1.667	1.108	26	5.712	4.815
13	1.890	1.151	27	5.910	5.741
14	2.059	1.185	28	6.229	6.566
15	2.314	1.270	29	6.675	7.493
16	2.765	1.365	30	7.003	9.226

4 Applications in economies with incomplete asset markets

The theory of vector sublattices and lattice-subspaces has been extensively used in the last years in Mathematical Economics, especially in the areas of incomplete markets and portfolio insurance. Recently, in [10], a new approach to the problem of completion by options of a two-period security market was introduced, which used the theory of positive bases in vector sublattices of \mathbb{R}^k (as described in [13]). Also, if x_1, x_2, \dots, x_n denotes a collection of linearly independent, positive vectors, of \mathbb{R}^k then if $X = [x_1, x_2, \dots, x_n]$ is a lattice subspace or, in the contrary case, if there exists a minimal lattice-subspace containing X then a solution to a cost minimization problem known as minimum-cost portfolio insurance always exists. In this section we shall discuss these interconnections and briefly describe the theoretical background on this subject. A full theoretical development is available in [2,3,4,7,8,10,11,12,13]. We will be also presenting how one can use the proposed functions, SUBlat and MINlat, in order to determine the completion of security markets as well as the solution of the minimum-cost portfolio insurance problem.

4.1 Completion of security markets

The problem of completion by options of a two-period security market in which the space of marketed securities is a subspace of \mathbb{R}^k has been studied in [10]. The present study involves vector sublattices generated by a subset B of \mathbb{R}^k of positive, linearly independent vectors, so we shall provide a computational solution to this problem by using the SUBlat function in order to provide the generated vector

sublattice Y as well as a positive basis for Y .

Let us assume that in the beginning of a time period there are n securities traded in a market. Let $\mathcal{S} = \{1, \dots, k\}$ denote a finite set of states and $x_j \in \mathbb{R}_+^k$ be the payoff vector of security j in k states. The payoffs x_1, x_2, \dots, x_n are assumed linearly independent so that there are no redundant securities. If $\theta = (\theta_1, \theta_2, \dots, \theta_n) \in \mathbb{R}^n$ is a non-zero portfolio then its payoff is the vector

$$T(\theta) = \sum_{i=1}^n \theta_i x_i.$$

The set of payoffs of all portfolios is referred as the space of *marketed securities* and it is the linear span of the payoffs vectors x_1, x_2, \dots, x_n in \mathbb{R}^k which we shall denote it by X , i.e.,

$$X = [x_1, x_2, \dots, x_n].$$

For any $x, u \in \mathbb{R}^k$ and any real number a the vector $c_u(x, a) = (x - au)^+$ is the *call option* and $p_u(x, a) = (au - x)^+$ is the *put option* of x with respect to the *strike vector* u and *exercise price* a .

Following the terminology of [10], let U be a fixed subspace of \mathbb{R}^k which is called *strike subspace* and the elements of U are the *strike vectors*. Then, the *completion by options* of the subspace X with respect to U is the space $F_U(X)$ which is defined inductively as follows:

- X_1 is the subspace of \mathbb{R}^k generated by \mathcal{O}_1 , where $\mathcal{O}_1 = \{c_u(x, a) | x \in X, u \in U, a \in \mathbb{R}\}$, denotes the set of call options written on the elements of X ,
- X_n is the subspace of \mathbb{R}^k generated by \mathcal{O}_n , where $\mathcal{O}_n = \{c_u(x, a) | x \in X_{n-1}, u \in U, a \in \mathbb{R}\}$, denotes the set of call options written on the elements of X_{n-1} ,
- $F_U(X) = \cup_{n=1}^{\infty} X_n$.

The completion by options $F_U(X)$ of X with respect to U is the vector sublattice of \mathbb{R}^k generated by the subspace $Y = X \cup U$. The details are presented in the next theorem,

Theorem 6 [10, Theorem 3] *In the above notation, we have*

- (i) $Y \subseteq X_1$,
- (ii) $F_U(X)$ is the sublattice $S(Y)$ of \mathbb{R}^k generated by Y , and
- (iii) if $U \subseteq X$, then $F_U(X)$ is the sublattice of \mathbb{R}^k generated by X .

Definition 7 [10, Definition 10] *Any set $\{y_1, y_2, \dots, y_r\}$ of linearly independent positive vectors of \mathbb{R}^k such that $F_U(X)$ is the sublattice of \mathbb{R}^k generated by $\{y_1, y_2, \dots, y_r\}$ is a **basic set** of the market.*

Theorem 8 [10, Theorem 11] *Any maximal subset $\{y_1, y_2, \dots, y_r\}$ of linearly independent vectors of \mathcal{A} is a basic set of the market, where $\mathcal{A} = \{x_1^+, x_1^-, \dots, x_n^+, x_n^-\}$,*

if $U \subseteq X$ and $\mathcal{A} = \{x_1^+, x_1^-, \dots, x_n^+, x_n^-, u_1^+, u_1^-, \dots, u_d^+, u_d^-\}$, if $U \subsetneq X$

Definition 9 [10, Definition 12] The space of marketed securities X is **complete by options** with respect to U if $X = F_U(X)$.

From theorem 2 and definition 9 it follows,

Theorem 10 [10, Theorem 13] The space X of marketed securities is complete by options with respect to U if and only if $U \subseteq X$ and $\text{card}R(\beta) = n$.

Theorem 11 [10, Theorem 14] The dimension of $F_U(X)$ is equal to the cardinal number of $R(\beta)$. Therefore, $F_U(X) = \mathbb{R}^k$ if and only if $\text{card}R(\beta) = k$.

In order to apply our method to the problem of completion of security markets we present the following example which is formerly featured in [10].

Example 12 [10, Example 16] Suppose that in a security market, the payoff space is \mathbb{R}^{12} and the primitive securities are:

$$x_1 = (1, 2, 2, -1, 1, -2, -1, -3, 0, 0, 0, 0)$$

$$x_2 = (0, 2, 0, 0, 1, 2, 0, 3, -1, -1, -1, -2)$$

$$x_3 = (1, 2, 2, 0, 1, 0, 0, 0, -1, -1, -1, -2)$$

and that the strike subspace is the vector subspace U generated by the vector

$$u = (1, 2, 2, 1, 1, 2, 1, 3, -1, -1, -1, -2).$$

Then, a maximal subset of linearly independent vectors of $\{x_1^+, x_1^-, x_2^+, x_2^-, x_3^+, x_3^-, u_1^+, u_1^-\}$ can be calculated by using the following code¹:

```
>>XX = [max(X,zeros(size(X)));max(-X,zeros(size(X)))];  
>>S = rref(XX');  
>>[I,J] = find(S);  
>>Linearindep = accumarray(I,J,[rank(XX),1],@min)';  
>>W = XX(Linearindep,:);
```

where X denotes a matrix whose rows are the vectors x_1, x_2, x_3, u . We can determine the completion by options of X i.e., the space $F_U(X)$, with the SUBlat function by using the following code:

```
>>[VectorSublattice,Positivebasis]=SUBlat(W')
```

The results then are as follows

¹ The rref function is a Matlab function that produces the reduced row echelon form of a given matrix by using Gauss Jordan elimination with partial pivoting (cf. [5]).

```

VectorSublattice =
1   2   2   0   1   0   0   0   0   0   0   0
0   2   0   0   1   2   0   3   0   0   0   0
1   2   2   1   1   2   1   3   0   0   0   0
0   0   0   0   0   0   0   0   1   1   1   2
2   0   4   0   0   0   0   0   0   0   0   0

Positivebasis =
0   0   0   0   0   0   0   0   1   1   1   2
0   0   0   1   0   0   1   0   0   0   0   0
0   0   0   0   0   4   0   6   0   0   0   0
4   0   8   0   0   0   0   0   0   0   0   0
0   6   0   0   3   0   0   0   0   0   0   0

```

4.2 Minimum cost portfolio insurance

In this section we shall, briefly discuss an investment strategy called minimum-cost portfolio insurance as a solution of a cost minimization problem. We will be also presenting how one can use the previous results for the calculation of the minimal lattice-subspace, in order to calculate the minimum-cost insured portfolio. In our model we use a method of comparing portfolios called portfolio dominance ordering. This ordering compares portfolios by means of the ordering of their pay-offs. Under this consideration we are able to use the order structure of the payoff space together with the theory of lattice-subspaces. In what follows we shall use the notation introduced in [11].

The model of security markets we study here is extended over two periods, namely period 0 and period 1. We assume n securities labeled by the natural numbers $1, 2, \dots, n$, acquired during the period 0 and that these n securities are described by their payoffs at date 1. The payoff of the i th security is in general a positive element x_i of an ordered space E which is called *payoff space*. In addition, we assume that the payoffs x_1, x_2, \dots, x_n are linearly independent so that there are no redundant securities and that the securities have limited liability which ensures the positivity of x_1, x_2, \dots, x_n . In [11], it is assumed that E is the space of real valued continuous functions $C(\Omega)$ defined in a compact, Hausdorff topological space Ω . A portfolio is a vector $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ of \mathbb{R}^n where θ_i is the number of shares of the i th security. The space \mathbb{R}^n is then known as *portfolio space*. If $\theta = (\theta_1, \theta_2, \dots, \theta_n) \in \mathbb{R}^n$ is a non-zero portfolio then its payoff is the vector

$$R(\theta) = \sum_{i=1}^n \theta_i x_i \in C(\Omega).$$

The operator R is one-to-one and is called the *payoff operator*. The pointwise ordering in $C(\Omega)$, induces the partial ordering \geq_R in the portfolio space \mathbb{R}^n and is defined as follows: For each $\theta, \phi \in \mathbb{R}^n$ we have

$$\theta \geq_R \phi, \text{ if and only if } R(\theta) \geq R(\phi).$$

This ordering is known as the *portfolio dominance ordering*. The set of payoffs of all portfolios, or the range space of the payoff operator, is the linear span of the payoffs vectors x_1, x_2, \dots, x_n in $C(\Omega)$ which we shall denote it by \mathcal{M} , i.e.,

$$\mathcal{M} = [x_1, x_2, \dots, x_n].$$

The subspace \mathcal{M} of $C(\Omega)$ is called the *asset span* of securities or the *space of marketed securities*.

Let us assume that $p = (p_1, p_2, \dots, p_n) \in \mathbb{R}^k$ is a vector of security prices and θ, ϕ are two portfolios. Then, the insured payoff on the portfolio $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ at the "floor" ϕ and in the price p is the contingent claim $R(\theta) \vee R(\phi)$.

The solution of the following cost minimization problem is referred to as the *minimum-cost insured portfolio*, or a *minimum-cost insurance of the portfolio θ at the floor ϕ and in the price p* ,

$$\min_{\eta \in \mathbb{R}^k} p \cdot \eta$$

subject to

$$R(\eta) \geq R(\theta) \vee R(\phi).$$

In [11] it is proved that if the payoff space is contained in a minimal lattice-subspace of $C(\Omega)$ then a minimum-cost insurance of the portfolio θ always exists. The details are included in the next theorem:

Theorem 13 [11, Theorem 15] *If the payoff space X is contained in a finite-dimensional minimal lattice-subspace Y of $C(\Omega)$ and the sum of the payoff vectors x_i is strictly positive, then a minimum-cost insurance of the portfolio θ at the floor ϕ and in the price p , exists and it is determined by solving the corresponding minimization problem.*

In order to apply our method, `MINlat`, we consider that $\Omega = \{1, 2, \dots, n\}$. Then, it is evident that $C(\Omega) = \mathbb{R}^n$. Therefore, in view of Theorem 13 and by using the `MINlat` function we are able to determine the minimum-cost insurance of the given portfolio θ at the floor ϕ and in the price p .

5 Conclusions

In this paper, new computational methods in order to determine vector sublattices and minimal lattice-subspaces of \mathbb{R}^k are presented. In order to reach our goal the study of a vector-valued function β is further involved by introducing two Matlab functions, namely `SUBlat`, and `MINlat`. The results of this work can give us an important tool in order to study the interesting problems of completion by options

of a two-period security market in which the space of marketed securities is a subspace of \mathbb{R}^k and in portfolio insurance. The experiment results, in subsection 3.2, show that our algorithm performs well. Also, note that the algorithm 2.3 determines the exact solution to the problems of completion by options and the minimum-cost insured portfolio. Finally we are convinced that, from the mathematical point of view, the proposed algorithm can be further analyzed independently, in terms of formal numerical analysis.

6 Appendix

The SUBlat function

```
function [Sublattice,Positivebasis] = SUBlat(B)
%SUBlat(B) provides the vector sublattice generated by
%a given finite collection of positive, linearly
%independent vectors of  $\mathbb{R}^n$ 
%B denotes the matrix whose columns are the given vectors
[N,M] = size(B);
Id = eye(N);
for i = 1:N,
    if norm(B(i,:),1)~=0,
Test(i,:) = 1/norm(B(i,:),1)*B(i,:);
    end
end
Matrix = Test;
[BB,m,n] = unique(Matrix,'rows');
Index = 1:N;
S = rref(BB');
[I,J] = find(S);
Linearindep = accumarray(I,J,[rank(BB),1],@min)';
mm = length(m);
nn = length(n);
Index1 = 1 : mm;
Index2 = setdiff(Index1,Linearindep);
YY = sum(B,2)';
TTT = setdiff(Index,m(Linearindep));
KK = Id(TTT,:);
TT = YY(1,TTT)';
T = diag(TT)*KK;
K = zeros(N);
K(TTT,:) = T;
Vec = zeros(mm-M,N);
if mm < nn,
```

```

for i = 1:length(Index2),
    DD = strmatch(Index2(i),n,'exact')' ;
    R = length(DD);
    if R >= 2,
        Vector = sum(K(DD,:));
    else
        Vector = K(DD,:);
    end
    Vec(i,:) = Vector;
end
[a,b] = find(Vec);
Vectors = Vec(unique(a),:);
Sublattice = [B';Vectors];
Positivebasis = SUBlatSUB(Sublattice');
else
    KKK = unique(K,'rows');
    [II,JJ] = find(KKK);
    Vectors = KKK(unique(II),:);
    Sublattice = [B';Vectors];
    Positivebasis = SUBlatSUB(Sublattice');
end
end

```

The MINlat function

```

function [MLS,Positivebasis] = MINlat(B)
%MINlat(B) provides the minimal lattice-subspace
%generated by a given finite collection of positive,
%linearly independent vectors of  $R^n$ 
%B denotes the matrix whose columns are the given vectors
[N,M] = size(B);
for i=1:N,
    if norm(B(i,:),1)~=0,
        Test(i,:) = 1/norm(B(i,:),1)*B(i,:);
    end
end
Matrix = Test;
[ii,jj] = find(Matrix);
Matrix1 = Matrix(unique(ii),:);
BB = unique(Matrix1,'rows');
M1 = rank(bsxfun(@minus,BB,BB(1,:)));
if M1<M,
    Utrans = bsxfun(@minus,BB,BB(1,:));
    Rot = orth(Utrans');
    Uproj = Utrans*Rot;
    Tri = convhulln(Uproj);
end

```

```

    VIndex = unique(Tri(:));
    P = BB(VIndex,:)' ;
    Q = length(VIndex);
else
    VIndex = unique(convhulln(BB));
    P = BB(VIndex,:)' ;
    Q = length(VIndex);
end
    Test = zeros(N,M);
for i=1:N,
    Test(i,:) = 1/norm(B(i,:),1)*B(i,:);
end
    Matrix = Test;
    BBB=Matrix';
    R = Q-M;
    Sol = zeros(Q,R);
for i = 1:N,
    Sol(:,i) = lsqnonneg(P,BBB(:,i));
end
    Solutions = Sol;
    Norms = sum(B,2)';
    Test1 = zeros(R,N);
for i = 1:R,
    Index1 = M+i;
    D = Solutions(Index1,:).*Norms*eye(N);
    Test1(i,:) = D;
end
    Minlatsub = [B';Test1];
    MLS = Minlatsub;
    Positivebasis = SUBlatSUB(MLS');

```

The SUBlatSUB function

```

function [positivebasis,dimensions] = SUBlatSUB(A)
%SUBlatSUB(A) provides the vector sublattice
%or the lattice-subspace of a given finite collection
%of positive, linearly independent vectors of  $\mathbb{R}^n$ 
%A denotes the matrix whose columns are the given vectors
if any(any(A<0))~=0,
    error('the initial matrix must have positive elements')
end
[N,M] = size(A);
if rank(A)~=M,
    error('the given vectors are linearly dependent')
end

```

```

for i=1:N,
    if norm(A(i,:),1)~=0,
        Test(i,:) = 1/norm(A(i,:),1)*A(i,:);
    end
end
matrix = Test;
[ii,jj] = find(matrix);
matrix1 = matrix(unique(ii),:);
u = unique(matrix1,'rows');
m = length(u(:,1));
if M == m,
    disp('vector sublattice')
    positivebasis = inv(u')*A';
    dimensions = [M m N]';
else
    m1 = rank(bsxfun(@minus,u,u(1,:)));
    if m1<M,
        utrans = bsxfun(@minus,u,u(1,:));
        rot = orth(utrans');
        uproj = utrans*rot;
        tri = convhulln(uproj);
        d = length(unique(tri(:)));
        if d == M,
            basis = inv(u(unique(tri(:)),:))'*A';
            disp('lattice-subspace')
            positivebasis = basis;
            dimensions = [M m d N]';
        else
            disp('not a lattice-subspace')
            dimensions = [M m d N]';
            positivebasis=[];
        end
    end
end
end

```

The testSUBlat function

```

function testSUBlat = testSUBlat(k,j)
%k is the the dimension of the Euclidean space R^k
%j is the number of the tested matrices
for i = 1:j,
    A = rand(k+2,k);
    testSUBlat = SUBlat(A);
end

```

The testMINlat function

```
function testMINlat = testMINlat(k,j)
%k is the the dimension of the Euclidean space  $\mathbb{R}^k$ 
%j is the number of the tested matrices
for i = 1:j,
    A = rand(k+2,k);
    testMINlat = MINlat(A);
end
```

Acknowledgments

The research of the first author was financially supported by the State Scholarship Foundation (IKY) in his postdoctoral studies.

References

- [1] Y.A. Abramovich, C.D. Aliprantis and I.A. Polyrakis, Lattice-Subspaces and positive projections, *Proc.R.Ir.Acad.*, **94A**(1994), 237-253.
- [2] C.D. Aliprantis, D.J. Brown and J. Werner, Incomplete derivative markets and portfolio insurance, *Cowles Foundation Discussion Paper*, **1126R**(1997), 1-13.
- [3] C.D. Aliprantis, D.J. Brown and J. Werner, Minimum-cost portfolio insurance, *Journal of Economic Dynamics & Control*, **24**(2000), 1703-1719.
- [4] C.D. Aliprantis, I.A. Polyrakis and R. Tourky, The cheapest hedge, *Journal of Mathematical Economics*, **37**(2002), 269-295.
- [5] MATLAB v 7.4(R2007b), Help Browser.
- [6] MATLAB User's Guide, The MathWorks Inc.
- [7] V.N. Katsikis, Computational methods in portfolio insurance, *Applied Mathematics and Computation*, **189**(2007), 9-22.
- [8] V.N. Katsikis, Computational methods in lattice-subspaces of $C[a, b]$ with applications in portfolio insurance, *Applied Mathematics and Computation*, **200**(2008), 204-219.
- [9] V.N. Katsikis, A Matlab-based rapid method for computing lattice-subspaces and vector sublattices of \mathbb{R}^n : Applications in portfolio insurance, *Applied Mathematics and Computation*, **215**(2009) 961972.
- [10] C. Kountzakis, I.A. Polyrakis, The completion of security markets, *Decisions in Economics and Finance*, **29**(2006),1-21.
- [11] I.A. Polyrakis, Linear Optimization in $C(\Omega)$ and Portfolio Insurance, *Optimization* **52**(2003), 221-239.
- [12] I.A. Polyrakis, Finite-dimensional lattice-subspaces of $C(\Omega)$ and curves of \mathbb{R}^n , *Transactions of the American Mathematical Society* **348**(1996), 2793-2810.

- [13] I.A. Polyakis, Minimal lattice-subspaces, *Transactions of the American Mathematical Society* **351**(1999), 4183-4203.